

Active Consensus over Sensor Networks via Randomized Communication

Lei Chen, Jeff Frolik

Abstract—Distributed consensus has been widely studied for sensor network applications. Whereas the asymptotic convergence rate has been extensively explored in prior work, other important and practical issues, including energy efficiency and link reliability, have received relatively little attention. In this paper, we present a distributed consensus approach that can achieve a good balance between convergence rate and energy efficiency.

The approach selects a subset of links that significantly contribute to the formation of consensus at each iteration, thus adapting the network's topology dynamically to the changes of the sensor states. A global optimization problem is formulated for optimal link selection, which is subsequently factorized into sub-problems that can be solved locally, and practically via approximation. An algorithm is derived to solve the approximation efficiently, using quadratic programming (QP) relaxation and random sampling. Simulations on networks of different types demonstrate that the proposed method reduces the communication energy costs without significantly impacting the convergence rate and that the approach is robust to link failures.

Index Terms—distributed average consensus, quadratic programming, convergence rate, energy efficiency, wireless sensor network.

I. INTRODUCTION

THE past decade has witnessed the growing use of wireless sensor networks (WSNs) in both industrial and scientific applications, including environment monitoring, structural monitoring, and emergence detection. With these networks in place, there is increasing interest in developing fast, robust, and energy efficient means to collect and aggregate data for both decision and control purposes. Collectively this area of research is called consensus forming or sensor fusion and can be roughly categorized as being either centralized or distributed among the sensor nodes. Distributed consensus, the focus of the work herein, has the advantages of improved robustness, scalability, and energy efficiency.

Much of the work to date in distributed consensus has been focused on algorithms and their convergence behavior. Xiao and Boyd [1] considered distributed consensus through linear iterations. Boyd [2] extended the work to randomized gossip. In both approaches, the convergence rate was shown to be closely related to the spectral structure of the network, i.e., the eigen-system of the network topology. Franceschelli *et al.* [3] proposed a new decentralized gossip algorithm based on broadcasts. Salehi and Jadbabaie [4] studied the asymptotic behavior of the consensus algorithms over switching graph. Recently, Chen *et al.* [5], [6] studied a new formulation for distributed consensus based on state derivative, they also proposed an efficient approach which allows each node in a network to individually determine whether consensus has

been attained. Pereira [7] evaluated the convergence of the consensus algorithm over networks with random asymmetric topology. Hatime [8] investigated the impact of topological characteristics on consensus building in multiagent systems.

Guided by these convergence analyses, various methods have been developed to design the optimal network structure that can achieve fastest convergence. Xiao and Boyd [1], [9] developed an optimization algorithm that minimizes the second largest eigenvalue. Das and Mesbahi [10] described a linear estimation algorithm based on distributed consensus by exploiting the clustering structure. Jin and Murray [11] proposed a new method that introduces virtual links between skeleton nodes to speed up convergence. Jakovetic *et al.* [12] presented a new formulation that addresses correlated random topologies. Jafarizadeh *et al.* [13] presented an analytical solution for the problem of fastest distributed consensus for a sensor network that is composed by two different symmetric star sensor networks. The previous work mentioned above focuses on theoretical analysis, where important practical aspects (e.g., energy efficiency) are often neglected. In this work, we focus on improving the energy efficiency of the network communication, i.e., reducing the energy cost while maintaining a reasonable convergence performance, which is crucial for the efficacy of a practical sensor network.

Earlier work on distributed consensus [1], [9]–[11] also focused on a static networks, optimizing the updating weights solely based on the network topology. Recently, randomized algorithms [2], [7], [14] that can generate graphs with varying topologies have become increasingly popular. However, in such an algorithm, the dynamic changes of the network topology typically follows a stationary process, which is devised without utilizing run-time information, that is, the information used is only from the initial network design.

In this paper, we take a different approach, exploiting the information available in run-time, i.e., the states maintained by the sensor nodes, to adapt the network topology on the fly. We observed that the usefulness of the various network links for obtaining consensus largely depends on run-time status. For example, the communication between two nodes with disparate states is more useful than that between nodes with similar states. Motivated by this intuition, we formulate an energy-constrained optimization problem, which seeks the most useful subset of links for consensus forming at each iteration, under a constraint on energy costs. In addition, we derive a randomized algorithm that can approximately solve this global problem in a distributed manner, by decomposing it into locally solvable problems. The active utilization of run-time states to optimize the communication topology along with the consensus building process clearly distinguishes this work

from other schemes, and we thus call it *active consensus*. Furthermore, as a practical issue, the robustness to link failures are also studied. This work extends earlier developments along these lines [15] by considering new topologies and investigating the robustness of the approach to link failures.

It is worth reiterating that the primary goal of this work is to address the practical problem of increasing energy efficiency. To test the efficacy of the proposed scheme, we performed simulation under a variety of conditions. The experimental results demonstrated that this method can substantially reduce the overall energy consumption, without significantly degrading the convergence time.

The remainder of this paper is organized as follows. In Section II, we revisit the theoretical analysis of the consensus algorithm and reveal its relations to a distributed optimization problem. In Section III, we formulate an optimization approach that selects the appropriate communication links in run-time, and then develop a randomized approximation algorithm that can be implemented locally. In Section IV, we present the simulation results under different topologies and channel conditions, which demonstrate the energy efficiency improvement achieved by the proposed method. Finally, we conclude the work in Section V.

II. THEORY

In this section, we first formalize a communication model for a sensor network in Section II-A, and thereon analyze the process of distributed average consensus in Section II-B. Then, we discuss a different perspective in Section II-C, which relates the formation of consensus to a distributed optimization procedure. As we shall see in Section III, this relation provides theoretical insight to guide the design of an energy-efficient communication scheme.

A. Iterative Linear Updates

Consider a sensor network comprised of a set of nodes, denoted by V , and a set of symmetric links¹ between them, denoted by E . Each node maintains a state value, which is dynamically updated based on the information from neighboring nodes. In this work, we assume time is discretized into time steps, and all the nodes of the network talk to their neighbors simultaneously at each time step (i.e., synchronized communication). In addition, we focus on the peer-to-peer communications that are typical in distributed sensor networks, where each communication step is between two specific nodes. Let $x_v(t)$ denote the state value maintained by the node v at time t , which can be intuitively understood as the node's estimate of the consensus value. For each node $v \in V$, $x_v(0)$ is initialized to be the value measured by the corresponding sensor, which is then iteratively updated using a linear combination of the values received from the neighbors, as follows

$$x_v(t) = x_v(t-1) - \delta \sum_{u \in \mathcal{N}_v} (x_v(t-1) - x_u(t-1)). \quad (1)$$

¹A network is symmetric if each link of the network is bidirectional, meaning that either end of the link can receive information of the other.

Here, $\mathcal{N}_v = \{u : \{v, u\} \in E\}$ is the set of neighbors of v , and δ is the step size. As we shall see later, the asymptotic behavior of this updating process is largely determined by the *Laplacian matrix* \mathbf{L} , which is defined as follows

$$\mathbf{L}(u, v) = \begin{cases} d_v & (u = v), \\ -1 & (\{u, v\} \in E), \\ 0 & (\text{otherwise}). \end{cases} \quad (2)$$

Here, $d_v = |\mathcal{N}(v)|$ is the degree of v (i.e., the number of neighbors of v). With the Laplacian matrix \mathbf{L} , the updating formula in Eq. (1) can be written in a vector form as

$$\mathbf{x}(t) = (\mathbf{I} - \delta \mathbf{L})\mathbf{x}(t-1). \quad (3)$$

Here, $\mathbf{x}(t)$ is a n -dimensional vector composed of the states of all nodes at time t . For conciseness, we use $\mathbf{W}(\mathbf{L}, \delta) \triangleq \mathbf{I} - \delta \mathbf{L}$ to denote the update matrix, which has

$$\mathbf{W}(\mathbf{L}, \delta)\mathbf{e}_i = (\mathbf{I} - \delta \mathbf{L})\mathbf{e}_i = (1 - \delta \lambda_i(\mathbf{L}))\mathbf{e}_i. \quad (4)$$

This implies that $\mathbf{W}(\mathbf{L}, \delta)$ has eigenvalues $1 - \delta \lambda_1(\mathbf{L}), \dots, 1 - \delta \lambda_n(\mathbf{L})$, which are associated with the same eigenvectors as \mathbf{L} , namely $\mathbf{e}_1, \dots, \mathbf{e}_n$. In particular, since $\lambda_1(\mathbf{L}) = 0$ and $\mathbf{e}_1 = \mathbf{1}$, we have $\mathbf{W}\mathbf{1} = \mathbf{1}$.

B. Distributed Average Consensus

Next, we consider how the updating process evolves over time and show that under certain conditions, the states of all nodes will reach an average value, called the *consensus*. We note that this is the result that underpins the distributed average consensus methodology, which we review here so as to provide a theoretical foundation for further discussion.

Let S_1 denote the subspace spanned by $\mathbf{e}_1 = \mathbf{1}$, and $D \triangleq S_1^\perp$ denote its orthogonal complement, which is spanned by the remaining eigenvectors $\mathbf{e}_2, \dots, \mathbf{e}_n$. Clearly, each vector $\mathbf{x} \in D$ satisfies $\mathbf{1}^T \mathbf{x} = 0$. Then each state vector $\mathbf{x}(t)$ can be uniquely decomposed into a linear combination of two components: one in S_1 and the other in D , as follows

$$\mathbf{x}(t) = \mu(t)\mathbf{1} + P_D \mathbf{x}(t). \quad (5)$$

Here, $\mu(t)$ is the average of the values in $\mathbf{x}(t)$, and $P_D \mathbf{x}(t)$ the projection of $\mathbf{x}(t)$ onto D . The component $\mu(t)\mathbf{1}$ has the same value at all nodes, which we call the consensus component; while the other component $P_D \mathbf{x}(t)$ reflects the differences between nodes, which we call the difference component. Then at time $t+1$, the updated state vector becomes

$$\mathbf{x}(t+1) = \mathbf{W}(\mathbf{L}, \delta)\mathbf{x}(t) = \mu(t)\mathbf{1} + \mathbf{W}(\mathbf{L}, \delta)P_D \mathbf{x}(t). \quad (6)$$

Here, we utilize the fact that $\mathbf{W}(\mathbf{L}, \delta)\mathbf{1} = \mathbf{1}$. It can be easily verified that this constitutes an orthogonal decomposition of $\mathbf{x}(t+1)$ along S_1 and D . On the other hand, such decomposition of $\mathbf{x}(t+1)$ can be expressed as

$$\mathbf{x}(t+1) = \mu(t+1)\mathbf{1} + P_D \mathbf{x}(t+1). \quad (7)$$

Therefore, we have

$$\mu(t+1) = \mu(t), \quad (8)$$

$$P_D \mathbf{x}(t+1) = \mathbf{W}(\mathbf{L}, \delta)P_D \mathbf{x}(t). \quad (9)$$

This implies that the consensus component is fixed over time. Therefore, we can use $\mu \triangleq \mu(t)$ to indicate this fixed average value. Applying the formulas above recursively results in

$$\mathbf{x}(t) = \mu \mathbf{1} + \mathbf{W}^t(\mathbf{L}, \delta) P_D \mathbf{x}(0). \quad (10)$$

Here, the evolution of the difference component, i.e., $P_D \mathbf{x}(t)$, depends on the spectrum of $\mathbf{W}(\mathbf{L}, \delta)$. Specifically, we have

$$\begin{aligned} \|P_D \mathbf{x}(t)\| &= \|\mathbf{W}^t(\mathbf{L}, \delta) P_D \mathbf{x}(0)\| \\ &\leq \|\mathbf{W}(\mathbf{L}, \delta)\|_D^t \|P_D \mathbf{x}(0)\|. \end{aligned} \quad (11)$$

Here, $\|\mathbf{W}\|_D$ indicates the operator norm of \mathbf{W} w.r.t. the subspace D , i.e., the maximum magnitude of its eigenvalues. According to Eq. (4), this norm is related to the eigenvalues of \mathbf{L} , as

$$\|\mathbf{W}(\mathbf{L}, \delta)\|_D = \max\{|1 - \delta\lambda_2(\mathbf{L})|, |1 - \delta\lambda_n(\mathbf{L})|\}. \quad (12)$$

Clearly, when $\|\mathbf{W}(\mathbf{L}, \delta)\|_D < 1$, $P_D \mathbf{x}(t)$ attenuates to $\mathbf{0}$ as $t \rightarrow \infty$. Consequently, $\mathbf{x}(t)$ approaches $\mu \mathbf{1}$, the consensus status, in which all nodes have the same state μ . We summarize the analysis above in the following theorem.

Theorem 1. *The updating process described in Eq. (3) converges in norm to $\mu \mathbf{1}$ when $\|\mathbf{W}(\mathbf{L}, \delta)\|_D < 1$, or equivalently, when both $|1 - \delta\lambda_2(\mathbf{L})|$ and $|1 - \delta\lambda_n(\mathbf{L})|$ are less than 1. Here, μ is the average of the values in $\mathbf{x}(0)$.*

Furthermore, the convergence time (up to some precision) is proportional to the value of $1/\log(\|\mathbf{W}(\mathbf{L}, \delta)\|_D^{-1})$. Therefore, one can attain the fastest convergence by minimizing $\|\mathbf{W}(\mathbf{L}, \delta)\|_D$. Particularly, given \mathbf{L} , the optimal step size is

$$\hat{\delta} = \frac{2}{\lambda_2(\mathbf{L}) + \lambda_n(\mathbf{L})}. \quad (13)$$

We note that Boyd *et al.* [9] derived a similar result on optimal choice of δ .

C. An Optimization-Based Perspective

Given a state vector \mathbf{x} , the sum of squared differences between neighboring nodes can be expressed concisely with a Laplacian matrix, as

$$\frac{1}{2} \sum_v \sum_{u \in \mathcal{N}_v} (x_v - x_u)^2 = \mathbf{x}^T \mathbf{L} \mathbf{x}. \quad (14)$$

Suppose the condition given by Theorem 1 is satisfied, it is a natural intuition that the differences between neighbor states gradually diminish as the updating proceeds. The following theorem establishes this intuition rigorously as a fact, stating that the value of $\mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t)$ decreases as t increases.

Theorem 2. *Let $\mathbf{x}(t)$ be an updating process described in Eq. (3), then*

$$\mathbf{x}(t+1)^T \mathbf{L} \mathbf{x}(t+1) \leq \|\mathbf{W}(\mathbf{L}, \delta)\|_D^2 \cdot (\mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t)). \quad (15)$$

In particular, if the condition in Theorem 1 is satisfied, namely $\|\mathbf{W}(\mathbf{L}, \delta)\|_D < 1$, then $\mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t)$ is a decreasing sequence, provided that the difference component of $\mathbf{x}(0)$ is non-zero.

This theorem together with the fact that $\mu(t) \equiv \mu$ suggests that the updating process described in Eq. (3) is actually

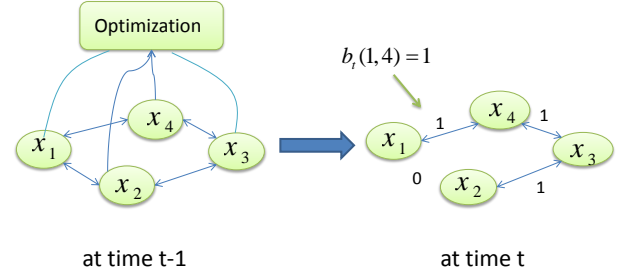


Fig. 1. This figure illustrates the proposed global optimization scheme on a network comprised of four nodes. At each iteration, previous states of the nodes are utilized as the input to the optimization problem given in Eq.(19), which finds the optimal subset of links that can decrease the differences between neighbors to the greatest extent, subject to the energy constraint. In this case, the optimal solution chooses three links: $\{a, b\}$, $\{b, c\}$, and $\{c, d\}$. They are used in the next iteration of updating.

minimizing the objective function $f_{\mathbf{L}}(\mathbf{x}) = \mathbf{x}^T \mathbf{L} \mathbf{x}$ in a distributed fashion, subject to the constraint $(\mathbf{1}^T \mathbf{x})/n = \mu$. This optimization problem has a unique optima $\hat{\mathbf{x}} = \mu \mathbf{1}$. Moreover, Eq. (15) provides the convergence rate of the objective value.

III. ENERGY-CONSTRAINED COMMUNICATION

Wireless sensor networks, especially those deployed on field, typically battery powered or utilize energy harvesting, and thus are usually subject to energy constraints. The primary goal of this work is to derive a practical communication scheme to improve the energy efficiency of distributed consensus. Inspired by the theoretical analysis in previous section, we present two approaches in this section. The basic idea underlying both methods is to adaptively choose a subset of links at each iteration, such that the information exchanged via these links contributes most to the formation of consensus. Specifically, in Section III-A, we first formalize the selection of links as an optimization problem and develop a methods that chooses the optimal subset of links by solving this problem. To make the algorithm practical, in Section III-B, we then factorize the optimization problem into sub-problems that can be solved locally using a randomized method.

A. Globally Optimal Link Selection

As discussed in Section II-C, building consensus over a network can be considered as minimizing the metric $\mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t)$. In this sense, we can measure the effectiveness of an updating iteration in terms of how much it decreases this metric. Suppose we only use a subset of links at time t to perform the update, then the updating formula can be written as

$$x_v(t) = x_v(t-1) - \delta \sum_{u \in \mathcal{N}_v} b_t(\{v, u\}) (x_v(t-1) - x_u(t-1)). \quad (16)$$

Here, $b_t(\{v, u\})$ indicates whether the link $\{v, u\}$ is used in the communication at time t , i.e. $b_t(\{v, u\}) = 1$ if this link is used and $b_t(\{v, u\}) = 0$ otherwise. Hence, $\mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t)$ depends on the choice of links, which can be captured by $\mathbf{b}_t = \mathbf{b}_t(e)$, where $e = \{u, v\} \in E$ the vector comprised of all the link indicators. Following this argument, we can choose the most effective subset of links, by finding the optimal \mathbf{b}_t in terms

of minimizing $\mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t)$. To derive a vector form of this objective function, we let $\mathbf{U}_{\mathbf{x}(t)}$ be an $|V| \times |E|$ matrix given by

$$\mathbf{U}_{\mathbf{x}(t)}(v, e) = \begin{cases} x_v(t) - x_u(t) & (\{v, u\} \in E) \\ 0 & (\text{otherwise}). \end{cases} \quad (17)$$

Then, Eq. (16) can be written into

$$\mathbf{x}(t) = \mathbf{x}(t-1) - \delta \mathbf{U}_{\mathbf{x}(t-1)} \mathbf{b}_t. \quad (18)$$

Hence, given $\mathbf{x}(t-1)$, the objective function of choosing the optimal subset of links can be written as

$$\begin{aligned} Q(\mathbf{b}_t) &= \frac{1}{2} \sum_{v \in V} \sum_{u \in \mathcal{N}_v} (x_v(t) - x_u(t))^2 = \mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t) \\ &= (\mathbf{x}(t-1) - \delta \mathbf{U}_{\mathbf{x}(t-1)} \mathbf{b}_t)^T \mathbf{L} (\mathbf{x}(t-1) - \delta \mathbf{U}_{\mathbf{x}(t-1)} \mathbf{b}_t) \\ &= \delta^2 \mathbf{b}_t^T \mathbf{H}_{\mathbf{x}(t-1)} \mathbf{b}_t - 2\delta \mathbf{f}_{\mathbf{x}(t-1)}^T \mathbf{b}_t + \text{const}. \end{aligned} \quad (19)$$

Here, we introduce $\mathbf{H}_{\mathbf{x}(t-1)}$ and $\mathbf{f}_{\mathbf{x}(t-1)}$ to simplify the notation, which are defined to be

$$\begin{aligned} \mathbf{H}_{\mathbf{x}(t-1)} &= \mathbf{U}_{\mathbf{x}(t-1)}^T \mathbf{L} \mathbf{U}_{\mathbf{x}(t-1)}, \\ \mathbf{f}_{\mathbf{x}(t-1)} &= \mathbf{U}_{\mathbf{x}(t-1)}^T \mathbf{L} \mathbf{x}(t-1). \end{aligned}$$

In general, using more links in the communication tends to speed up the convergence. As a result, purely pursuing fastest convergence would lead to the selection of many links, incurring high energy consumption. In practice, it is usually more desirable to seek a balance between convergence rate and the energy cost. To this end, we impose a cost constraint to this optimization problem. Particularly, we associate each link $e \in E$ with a cost value c_e , which reflects the energy needed to communicate via this link. Let $\mathbf{c} = (c_e)_{e \in E}$ be the vector composed of the cost values of all links. Then, the constraint can be expressed as

$$\mathbf{c}^T \mathbf{b}_t = \sum_{e \in E} c_e b_t(e) \leq C. \quad (20)$$

This means that the total communication cost at time t should not exceed C . Combining Eq. (19) and Eq. (20), we obtain a constrained optimization problem, given by

$$\text{minimize } \frac{1}{2} \delta^2 \mathbf{b}_t^T \mathbf{H}_{\mathbf{x}(t-1)} \mathbf{b}_t - \delta \mathbf{f}_{\mathbf{x}(t-1)}^T \mathbf{b}_t, \quad (21)$$

$$\text{s.t. } \mathbf{c}^T \mathbf{b}_t \leq C. \quad (22)$$

Note that each entry of \mathbf{b}_t can only take a value from $\{0, 1\}$. Hence, this is a binary integer programming problem². In general, finding the optimal solution to this problem is NP-hard. However, taking advantage of the quadratic form of the problem, we derive an efficient algorithm as follows. First, through a relaxation that allows the value of $b_t(e)$ to be any real number in $[0, 1]$, this problem reduces to a quadratic programming (QP) problem with linear constraints that can be readily solved. Denote the solution to the QP as \mathbf{p}_t . To turn this into a binary solution \mathbf{b}_t , we treat each value of $p_t(e)$ as a probability, and thus obtain the value of $b_t(e)$ by random sampling. Particularly, for each $e \in E$, we draw $b_t(e) \in \{0, 1\}$

with $P(b_t(e) = 1) = p_t(e)$. For instance, if $p_t(e) = 0.7$, then we turn on the edge e at time t with a chance of 70%. An illustration of the proposed global optimization scheme is shown in Fig 1.

We note that \mathbf{b}_t obtained using this approximate method is not necessarily the optimal solution to the original binary integer programming problem, and one can further refine this solution using MCMC [16] simulated annealing. However, we find that this is not necessary, as the solution \mathbf{b}_t derived via sampling is very close to the true optima in most cases. Suppose \mathbf{b}_t^* is the true optimal solution to the original problem, *i.e.* it attains the minimum objective among all binary vectors, then $Q(\mathbf{b}_t^*) \leq Q(\mathbf{b}_t)$. Moreover, \mathbf{p}_t is the optimal solution to the relaxed problem, *i.e.* it attains the minimum within a larger domain that allows real values, implying that $Q(\mathbf{p}_t) \leq Q(\mathbf{b}_t^*)$. Together, we have

$$Q(\mathbf{b}_t) \geq Q(\mathbf{b}_t^*) \geq Q(\mathbf{p}_t). \quad (23)$$

In experiments, we found that $Q(\mathbf{b}_t)$ is very close to $Q(\mathbf{p}_t)$ in most cases, implying that $Q(\mathbf{b}_t)$ is even more close to $Q(\mathbf{b}_t^*)$, as $Q(\mathbf{p}_t)$ provides a lower bound to the minimum of the original problem. This, in other words, means that $Q(\mathbf{b}_t)$ is close to the true optimum.

B. Locally Optimal Link Selection

As we shall see in next section, the algorithm described above is very effective in optimizing the network topology under cost constraints. However, the reliance on solving a QP problem over the joint state vector makes it impractical in a distributed context. To address this issue, we consider an approximation, with which, the optimization problem can be decomposed into a collection of sub-problems that can be solved locally by nodes.

Given $\mathbf{x}(t-1)$, we can rewrite the objective function $Q(\mathbf{b}_t)$ in Eq. (19) as follows.

$$Q(\mathbf{b}_t) = \sum_{v \in V} Q_v(\mathbf{b}_t), \quad (24)$$

with

$$Q_v(\mathbf{b}_t) = \frac{1}{2} \sum_{u \in \mathcal{N}_v} (x_v(t) - x_u(t))^2. \quad (25)$$

Here, $x_v(t)$, as given by Eq. (16), is the value of node v at next time step t , and therefore $Q_v(\mathbf{b}_t)$ is the sum of the squared differences between the value of v and those of its neighbors.

In this way, we decompose $Q(\mathbf{b}_t)$, the overall objective, into the sum of local objectives $Q_v(\mathbf{b}_t)$, each associated with a node v . Our goal here is to divide the original optimization problem into ones that can be solved separately and locally by the sensor nodes, *i.e.* each node can solve its corresponding problem solely based on the information that it has.

However, $Q_v(\mathbf{b}_t)$ cannot be optimized locally with its current form. Note that $Q_v(\mathbf{b}_t)$ depends on $x_u(t)$ for each $u \in \mathcal{N}_v$, which in itself depends on its neighbor values, as

$$x_u(t) = x_u(t-1) - \delta \sum_{w \in \mathcal{N}_u} b_t(\{u, w\})(x_u(t-1) - x_w(t-1)).$$

The issue here is that w is not necessarily v 's neighbor, and

²A binary integer programming problem is an optimization problem, of which each variable can only take a binary value, *i.e.* either 0 or 1.

for such nodes, node v has no idea of their values. To address this issue, we simply ignore non-neighbors of v , and derive an approximated updating formula below

$$\tilde{x}_{v:u}(t) = x_u(t-1) - \delta \sum_{w \in \mathcal{N}_u \cap \mathcal{N}_v} (x_w(t-1) - x_u(t-1)). \quad (26)$$

Here, $\mathcal{N}_u \cap \mathcal{N}_v$ is the set of neighbors shared by both u and v . The node v uses the values of these nodes to make an approximate prediction of what the value u is at time t , denoted by $\tilde{x}_{v:u}(t)$, resulting in the following approximate local objective:

$$\tilde{Q}_v(\mathbf{b}_t(E_v)) = \frac{1}{2} \sum_{u \in \mathcal{N}_v} (x_v(t) - \tilde{x}_{v:u}(t))^2. \quad (27)$$

Here, $E_v = \{(v, u) : u \in \mathcal{N}_v\}$ is the subset of edges incident with node v , and $\mathbf{b}_t(E_v)$ is the corresponding sub-vector \mathbf{b}_t , which consists of all the link indicators for v .

We also break the original total cost constraint $\mathbf{c}^T \mathbf{b}_t \leq C$ into local constraints as follows, so as to make the sub-problems locally solvable.

$$\sum_{e \in E_v} c_e b_t(e) \leq C_v. \quad (28)$$

Here, C_v is the cost bound individually imposed on node v , which is proportional to the degree of node v .

We have obtained a local optimization problem for each node v , i.e. to minimize \tilde{Q}_v given by Eq. (27), subject to the local constraint given by Eq. (28). This problem can be solved separately by the node v , without consulting other nodes. This is an integer programming problem, which is difficult to solve directly. Again, we relax the binary value $b_t(e) \in \{0, 1\}$ to a real value $p_t(e) \in [0, 1]$, thus turning it into a QP problem. Here, $p_t(e)$ can be considered as the probability that the link e should be used at time t . Note that for each edge $e = \{u, v\} \in E$, we obtain two probability values for e , respectively when optimizing Q_u and Q_v . These values are not the same in general. We take a simple method, using the average of them to be the probability of setting $b_t(e) = 1$.

IV. DEMONSTRATION OF APPROACH

To test the practical performance of the proposed approach, we conducted simulations on synthetic networks for both the globally and locally optimal link selection methods (respectively presented in Sections III-A and III-B).

In the simulation, we consider four types of networks: (1) *uniform-degree networks*, where all nodes have similar degrees, i.e. the number of neighbors; (2) *nonuniform-degree networks*, where a small portion of nodes have substantially higher degrees than others; (3) *star networks*, where a central node is connected to all the other nodes; and (4) *chain networks*, where all the nodes are connected in a pattern of chain.

For each network type, we randomly produced the network topology and conducted simulations under a variety of configurations. The step size, δ , was found using Eq. (13), which achieves the fastest asymptotic convergence rate when all the links are used in the iterative updates. The upper bound in

the cost constraint (Eqs. (20) and (28)) were set to different values in our test, so as to study the trade-off between energy cost and convergence rate with the proposed schemes.

The simulations were conducted as follows. First, an initial state value $x_v(0)$ was independently drawn from a standard normal distribution ($\sigma^2 = 1$) for each node of the network. With the initialized states, we ran the iterative updating procedure, using the adaptive scheme to choose a subset of links at each iteration. The updating procedure was stopped when the difference between the highest and lowest state values were below a tolerance value ε , as

$$\|\max(\mathbf{x}(t)) - \min(\mathbf{x}(t))\| < \varepsilon \quad (29)$$

For this work, the tolerance ε was set to 10^{-3} . Along with the simulation of an adaptive scheme, we also ran a simulation upon the same network and same initial states without adaptation (using all links at every iteration) to establish the baseline for comparison. Let m be the number of edges. Then for global optimization, the value C in the cost constraint (see Eq. (20)) is set to αm . The critical parameter in our work is α which is the ratio of the maximum number of links allowed at each iteration to the total number of available links. Similarly, the value C_v for local optimization (see Eq. (28)) is set to αd_v , where d_v is the degree of node v .

We consider two different metrics to assess the performance of the selective communication scheme: (1) *Cost*: the total communication cost, i.e., the sum of the costs of all iteration (the cost of each iteration is defined to be the number of edges used in that iteration); (2) *Time*: the number of iterations that is needed to attain consensus. We also evaluate the ratios of the time and cost obtained using the selective schemes to that obtained by the baseline using all available links at each iteration, in order to quantitatively measure how link selection influences the performance. The cost ratio measures how much the selective scheme improves the energy efficiency under the given setting, while the time ratio reflects how much the use of the selective scheme influences the convergence rate.

A. Simulation on Uniform-Degree Networks

We first studied the proposed algorithms on uniform-degree networks. As illustrated in Fig 2(a), a uniform-degree network used in this simulation contains $n = 100$ nodes, each connecting to five nodes on average.

We tested both globally optimal and locally optimal selection schemes on uniform-degree networks. Fig 2(b) and Fig 2(c) respectively shows the cost ratios and the time ratios obtained with different α values, and different degrees d . From the results, we observe the following. First, both schemes can achieve substantial improvement on energy efficiency, with the convergence time maintained at a reasonable level. Second, the cost ratio decreases and the time ratio increases, as the value of α decreases. For example, under the setting with $d = 20$, when α is reduced from 0.8 to 0.3, the cost ratio obtained with global optimization decreases from 0.4 to 0.3, while the time ratio increases from below 1.0 to 1.2. With local optimization scheme, the cost ratio decreases from 0.7 to 0.45, while the time ratio increases from 1.5 to 2.0. Third,

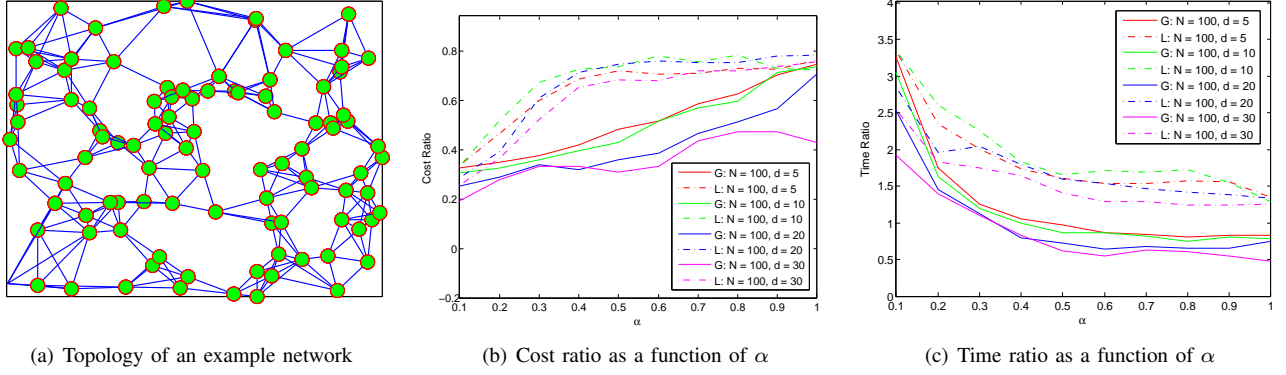


Fig. 2. (a) shows an example of a uniform-degree network. (b) and (c) show the simulation performance obtained on uniform-degree networks with varying values of α , respectively in terms of cost ratios and time ratios. Here, solid and dash-dot lines represent the results obtained by the selective communication scheme using global optimization and local optimization respectively. The different curves represent the results with different degrees.

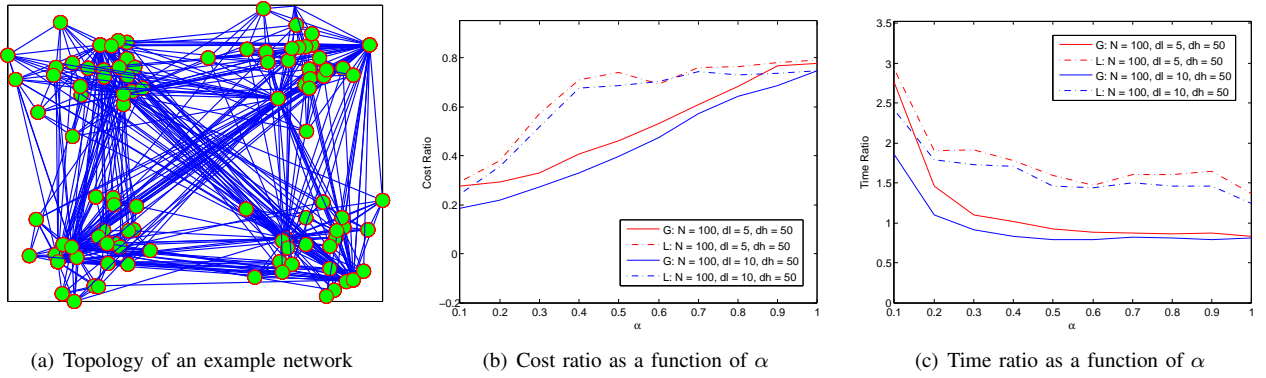


Fig. 3. (a) shows an example of a nonuniform-degree network. (b) and (c) show the simulation performance obtained on uniform-degree networks with varying values of α , respectively in terms of cost ratios and time ratios. Here, solid and dash-dot lines represent the results obtained by the selective communication scheme using global optimization and local optimization respectively. The different curves represent the results with different degrees.

greater improvement is achieved on the network with higher degrees, i.e., each node has more neighbors. Intuitively, in such networks, edges can be more redundant. As a consequence, we can suppress more edges at each iteration, without noticeably affecting the convergence.

B. Simulation on Nonuniform-Degree Networks

Next, we performed simulations on another type of networks, the non-uniform degree networks, to see whether the proposed methods exhibit different behaviors.

As shown in Fig 3, a non-uniform network is composed of four clusters of nodes, each with 25 nodes. Each cluster has two high-degree nodes that link to 50 other nodes, among which 24 are within the same cluster, and 26 are in others. Other nodes have much lower degree (5 on average) and only connect to nodes in the same cluster. Such networks can be considered as simplified versions of the hierarchical networks, where the high-degree nodes play a crucial role for both within-cluster consensus and cross-cluster state propagation. Thus the selection of the links that are incident with such nodes is important.

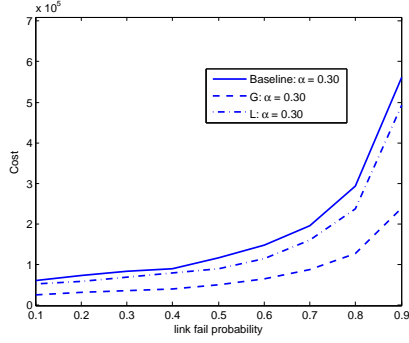
Fig 3(b) and Fig 3(c) respectively show the cost ratios and time ratios, obtained with different α values. Again, both adaptive schemes (global optimization and local optimization) yield

significant improvement on energy efficiency. For example, when $\alpha = 0.3$, the scheme using global optimization saves about 70% of the communication cost, with slightly increased convergence time (the time ratio is 1.0 – 1.2). Under the same setting, the scheme with local optimization can save about 60% of the energy, while the convergence time is still maintained at a reasonable level (time ratio is about 1.7). In addition, we observe similar trends from the results, e.g. the cost ratio decreases and the time ratio increases as α increases.

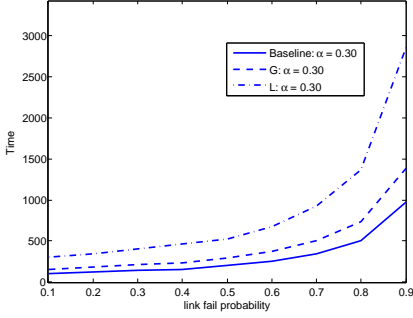
C. Performance with Link Failures

We also investigate how the algorithms work in the presence of link failures. Sensor networks, especially those deployed in field, are often subject to communication failures, due to various causes, such as adversarial environmental changes, channel interference, etc. Therefore, the practical effectiveness of a communication scheme is influenced, to a great extent, by its robustness against such problems.

We study this issue via a simulation under a simplified setting, where we assume that each communication link fails independently with a certain probability, we randomly turned off some links at each communication step. We ran the simulations on a uniform degree network of $d = 10$ similar



(a) Cost as a function of link fail probability p



(b) Time as a function of link fail probability p

Fig. 4. The performances obtained with varying link failure probability. Here, solid, dashed, and dash-dot lines respectively represent the results obtained using baseline scheme (without link selection), global optimization, and local optimization.

to Fig 2(a) and α is set to 0.3, with different probabilities of failure, ranging from 0.1 to 0.9.

To take a more close examination of how the link failure affects performance, we plot the absolute cost and convergence time for both schemes, as shown in Fig 4, instead of their ratios. The results clearly show as expected that for all schemes, both communication cost and convergence time increases as the probability of link failure increases. However, regardless of the variation of failing probabilities, the proposed link selection schemes consistently achieve notable improvement on energy efficiency. Particularly, the scheme using global optimization can save over 60% of the total cost, while the one using local optimization can save over 20%. In addition, we observe that while the convergence time increases as the communication links become more likely to fail, it is still maintained at a reasonable level, as compared to the baseline. With these results, we contend the selective communication schemes are robust to link failures.

D. Simulation on Star and Chain Networks

Finally, we present a more detailed study on two special types of networks: star networks and chain networks. A star network consists of a central node, which is directly connected to all other nodes. A chain network consists of nodes arranged in form of a chain, with each node connected to at most two nodes. These two types of networks represent two topological extremes. In a star network, information is quickly exchanged

between center and leaf nodes, while in a chain network, information is propagated much more slowly, as it has to pass along the chain, from one end to the other. Fig 5(a) and Fig 6(a) respectively illustrate a star network with 50 nodes and a chain network with 10 nodes.

In this experiment, we considered star networks with different number of nodes, with an aim to study how network structure affects energy efficiency and network performance. Here, we focus on local scheme, as it is practically feasible. Empirically, we set α to 0.3 in the simulation, which means up to 30% of the total available links are allowed at each iteration.

The results obtained on star networks are shown in Fig 5. Specifically, Fig 5(b) and Fig 5(c) respectively show the total communication costs and the number of iterations needed to achieve convergence, both as functions of the network size n . Fig 6 shows the results on chain networks, in the same fashion.

The results of star and chain networks present similar trends. When the network size increases, the communication cost rises and it takes more time to reach consensus. More importantly, we also observe significant reduction of the total communication cost, with the use of local optimization scheme. Note that it takes longer to achieve convergence as compared to the baseline setting. This is not surprising, as only a small subset of links are activated at each communication step. In particular, when the network size is 50, about 35% and 55% of cost can be saved for star network and chain network respectively, and it takes 2.2 and 1.6 of the convergence time accordingly. It is worth emphasizing that the value of this work is that it provides an effective way for one to trade off convergence performance for energy efficiency. This is very useful under many practical circumstances, where efficient use of energy may be more important than achieving the optimal convergence time.

V. CONCLUSION

In this paper, we presented two approaches to improve the energy efficiency of distributed average consensus. Specifically, we revisited the analysis of the consensus process, and established it as a distributed optimization procedure that minimizes the value of $\mathbf{x}(t)^T \mathbf{L} \mathbf{x}(t)$. Motivated by this relation, we first developed an approach that chooses an optimal subset of links for communication at each iteration, by minimizing this value subject to a cost constraint. Then, via approximation, we factorized this problem into a set of sub-problems that can be solved locally, resulting in an approach that is practical in a distributed context. In solving this problems, we derive a simple yet effective method, which first solve a relaxed QP problem, and then obtain a subset of links by random sampling.

We performed simulations on various types of networks to test the performance of the proposed schemes, and compared results with a baseline scheme without link selection. The results acquired on uniform, non-uniform, star and chain networks clearly demonstrate that the proposed methods have consistently improved energy efficiency for the distributed consensus problem. Based on these results, we contend that

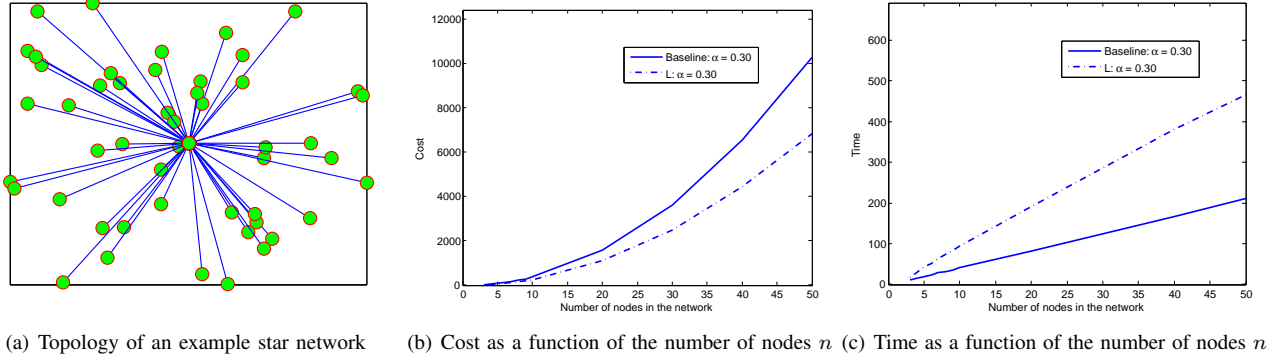


Fig. 5. (a) shows an example of a star network. (b) and (c) show the simulation performance obtained on star networks with varying number of nodes n , respectively in terms of cost and time. Here, solid and dash-dot lines respectively represent the results obtained using baseline scheme (without link selection) and local optimization.

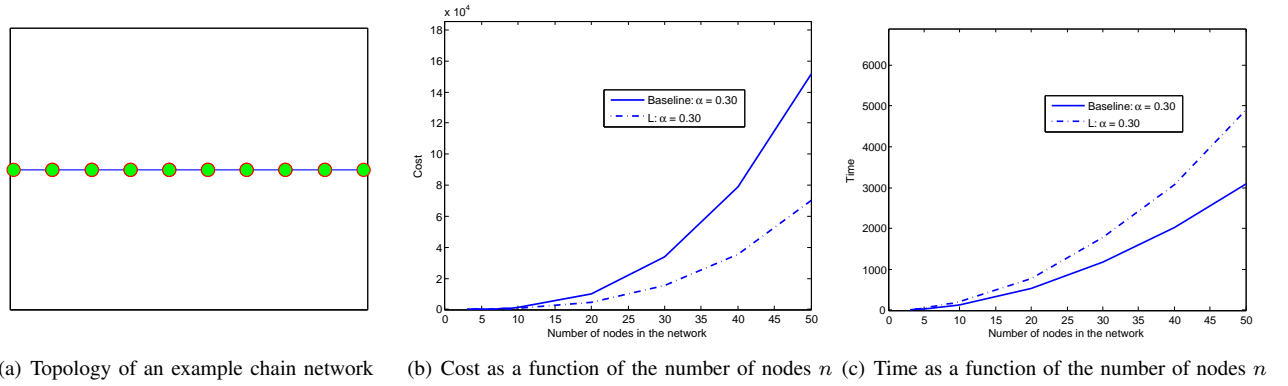


Fig. 6. (a) shows an example of a chain network. (b) and (c) show the simulation performance obtained on chain networks with varying number of nodes n , respectively in terms of cost and time. Here, solid and dash-dot lines respectively represent the results obtained using baseline scheme (without link selection) and local optimization.

dynamic link selection for distributed computing in sensor networks should be considered as a viable methodology to improve the energy efficiency of sensor networks. We also studied the influence of the link reliability on the performance of the proposed methods. Our results provide practical guidance as to how to choose design parameters in order to strike a balance between convergence speed and energy cost.

REFERENCES

- [1] X. Lin and S. Boyd, "Fast linear iterations for distributed averaging," in *Proc. of 42th IEEE Conference on Decision and Control*, 2003, pp. 4997 – 5002.
- [2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.
- [3] M. Franceschelli, A. Giua, and C. Seatzu, "Distributed averaging in sensor networks based on broadcast gossip algorithms," *Sensors Journal, IEEE*, vol. 11, no. 3, pp. 808–817, Mar. 2011.
- [4] A. T. Salehi and A. Jadbabaie, "Necessary and sufficient conditions for consensus over random independent and identically distributed switching graphs," in *Proc. of 46th IEEE Conference on Decision and Control*, 2007.
- [5] L. Chen, G. Carpenter, S. Greenberg, J. Frolik, and X. S. Wang, "An implementation of decentralized consensus building in sensor networks," *Sensors Journal, IEEE*, vol. 11, no. 99, pp. 667–675, Mar. 2011.
- [6] L. Chen and J. Frolik, "Distributed consensus building for sensor networks," *ISRN Communications and Networking*, September 2011.
- [7] S. S. Pereira, "Mean square convergence of consensus algorithms in random wsns," *IEEE Transactions on Signal Processing*, vol. 59, no. 9, p. 4501, 2011.
- [8] H. Hatime, "Impact of topological characteristics on consensus building in multiagent systems," *Sensors Journal, IEEE*, vol. 11, no. 12, pp. 3377 –3387, Dec 2011.
- [9] S. Boyd, P. Diaconis, and L. Xiao, "Fastest mixing markov chain on a graph," *SIAM REVIEW*, vol. 46, no. 4, pp. 667–689, 2003.
- [10] A. K. Das and M. Mesbahi, "Distributed linear parameter estimation in sensor networks based on laplacian dynamics consensus algorithm," in *Proc. 3rd Annual IEEE Communications Society on Sensor and Ad Hoc Communications and Networks, SECON*, 2006, pp. 440 –449.
- [11] Z. Jin and R. Murray, "Random consensus protocol in large-scale networks," in *Proc. of 46th IEEE Conference on Decision and Control*, 2007.
- [12] D. Jakovetic, J. Xavier, and J. M. F. Moura, "Consensus in correlated random topologies: Weights for finite time horizon," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP*, 2010, pp. 2974–2977.
- [13] S. Jafarizadeh and A. Jamalipour, "Fastest distributed consensus problem on fusion of two star sensor networks," *Sensors Journal, IEEE*, vol. 11, no. 10, pp. 2494 –2506, Oct 2011.
- [14] S. Kar and J. M. F. Moura, "Distributed consensus algorithms in sensor networks with imperfect communication: link failures and channel noise," *IEEE Transactions on Signal Processing*, vol. 57, no. 1, pp. 355 –369, 2009.
- [15] L. Chen and J. Frolik, "Active consensus over sensor networks via selective communication," *Ninth Annual IEEE Communication Society Conference on Sensor, Mesh, and Ad Hoc Communications and Networks (SECON 2012)*, June 2011.
- [16] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2010.